

Parallel inference for massive distributed spatial data using low-rank models

Matthias Katzfuss* Dorit Hammerling[†]

November 11, 2014

Abstract

Due to rapid data growth, statistical analysis of massive datasets often has to be carried out in a distributed fashion, either because several datasets stored in separate physical locations are all relevant to a given problem, or simply to achieve faster (parallel) computation through a divide-and-conquer scheme. In both cases, the challenge is to obtain valid inference based on all data without having to move the datasets to a central computing node. We show that for a very widely used class of spatial low-rank models, which can be written as a linear combination of spatial basis functions plus a fine-scale-variation component, parallel spatial inference and prediction for massive distributed data can be carried out exactly, meaning that the results are the same as for a traditional, non-distributed analysis. The computational cost of our distributed algorithms is linear in the number of data points, while the communication cost does not depend on the data sizes at all. After extending our results to the spatio-temporal case, we illustrate our methodology by carrying out distributed spatio-temporal particle filtering inference on total precipitable water measured by three different satellite sensor systems.

Key words: Distributed computing; Gaussian process; particle filter; predictive process; spatial random effects model; spatio-temporal statistics

1 Introduction

While data storage capacity and data generation have increased at a factor of thousands in the past decade, the data transfer rate has increased at a factor of less than ten (Zhang, 2013). It is therefore of increasing importance to develop analysis tools that minimize the movement of data and perform necessary computations in parallel where the data reside (e.g., Fuller and Millett, 2011). Here we consider two situations in which *distributed data* can arise:

*Department of Statistics, Texas A&M University, katzfuss@gmail.com

[†]Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research; Department of Statistics, University of Washington

Situation 1: Several massive datasets that are stored in separate data centers (servers) are all relevant to a given problem, and moving them to one central computing node for analysis is either too costly due to their large size or not desirable for other reasons such as avoiding unnecessary duplicated storage requirements. The goal then is to move the analysis to the data instead of the other way around (e.g., Shoshani et al., 2010).

Situation 2: All relevant data to a given problem are stored in one location, but a “divide-and-conquer” approach with several nodes working in parallel on different chunks of the data is necessary to achieve sufficiently fast computation. Especially if none of the available nodes can hold the quantities required for analysis of the entire dataset in working memory, a divide-and-conquer approach that avoids expensive repeated input/output can lead to substantial speed-ups.

The goal in both of these situations is to obtain valid inference based on all data at a number of computers or servers, without moving the individual datasets between servers. The focus in this article is on Situation 1, but all results are also applicable to Situation 2 without modification.

In the spatial and environmental sciences, both of the described distributed-data situations arise frequently. Because analysis of a spatial dataset of size n usually involves the data covariance matrix that has n^2 elements, Situation 2 applies to datasets of even moderate size. Situation 1 arises when several datasets containing information about a particular environmental variable are stored in different data centers throughout the US or the world, and we aim to obtain spatial inference and prediction based on all of them. For example, remotely sensed measurements of sea surface temperature are available both from the National Oceanic and Atmospheric Administration’s (NOAA’s) Advanced Very High Resolution Radiometer and from the National Aeronautics and Space Administration’s (NASA’s) Moderate Resolution Imaging Spectroradiometer, while measurements of column-integrated carbon dioxide are conducted by NASA’s Orbiting Carbon Observatory-2 and Atmospheric InfraRed Sounder, Japan’s Greenhouse Gases Observing Satellite, and other instruments. In this article, we will illustrate our methodology by making on-line spatio-temporal inference on a spatial variable called total precipitable water, based on measurements made by three major sensor systems stored at three associated data centers.

We consider here spatial low-rank models that consist of a component that can be written as a linear combination of spatial basis functions and a spatially independent fine-scale-variation term. Despite some recent criticism of their ability to approximate the likelihood of spatial processes with parametric covariances in certain low-noise situations (Stein, 2014), low-rank models are a very widely used class of models for large spatial datasets (see Section 2 below) because of their scalability for massive data sizes, and their predictive performance has been shown to compare favorably to other (full-rank) approaches (Bradley et al., 2014). Note that here we do not advocate for or propose a particular spatial low-rank model — rather, we are presenting distributed algorithms for inference that are applicable to all members of the class of spatial low-rank models.

Specifically, by writing spatial low-rank models in state-space form and applying the decentralized Kalman filter (originally developed by Rao et al., 1993, for tracking and surveillance), we show that basic inference for these models can be carried out *exactly* for massive

distributed spatial data, while only relying on (*parallel*) *local* computations at each server. The required number of floating-point operations (flops) is linear in the number of measurements, while the communication cost does not depend on the data size at all. Based on this main algorithm, we derive further algorithms for parameter inference and spatial prediction that are similarly well-suited for massive distributed data, and we extend the results to the spatio-temporal case. The results of our parallel distributed algorithms are exactly the same as those obtained by a traditional, non-distributed analysis with all data on one computational node, and so we do *not* ignore spatial dependence between the data at different servers.

General-purpose computer-science algorithms for massive distributed data are not well suited to the distributed-spatial-data problem described above, as solving the linear systems required for prediction and likelihood evaluation would involve considerable movement of data or intermediary results. In the engineering literature, there has been some work on distributed Kalman filters for spatial prediction based on measurements obtained by robotic sensors (Cortés, 2009; Xu and Choi, 2011; Graham and Cortés, 2012), but because the sensors are typically assumed to collect only one measurement at a time, we are not aware of any treatment of the case where the individual datasets are massive.

In the statistics literature, we are also not aware of previous treatment of the distributed-spatial-data problem of Situation 1, although it is possible to adapt some approaches proposed for analyzing (non-distributed) massive spatial data to the distributed case — which is what we are doing with low-rank models in this article. The most obvious other approach is to simply approximate the likelihood for parameter estimation by dividing the data into blocks and then treating the blocks as independent, where in the distributed context each block would correspond to one of the distributed datasets. However, in most applications the distributed datasets were not necessarily collected in distinct spatial regions, and so block-independence approaches might ignore significant dependence between different blocks if there is substantial overlap in spatial coverage of the blocks. While methods such as composite likelihoods (e.g., Vecchia, 1988; Curriero and Lele, 1999; Stein et al., 2004; Caragea and Smith, 2007, 2008; Bevilacqua et al., 2012; Eidsvik et al., 2014) have been proposed to allow for some dependence between blocks, it is not clear how well these methods would work in our context, and how spatial predictions at unobserved locations should be obtained (e.g., to which block does the prediction location belong?). Other efforts to implement parallel algorithms for large spatial datasets (e.g., Lemos and Sansó, 2009) also exploit being able to split the data by spatial subregions and hence might not be suitable to distributed data in Situation 1.

This article is organized as follows. We begin with a brief review of low-rank spatial models in Section 2. We then focus on the distributed-data setting, describing a basic parallel algorithm for inference (Section 3), discussing inference on model parameters and presenting important simplifications for fixed basis functions (Section 4), describing how to do spatial prediction (Section 5), and extending the methodology to the spatio-temporal setting (Section 6). In Section 7, we present an application to total precipitable water measured by three sensor systems, and we conclude in Section 8.

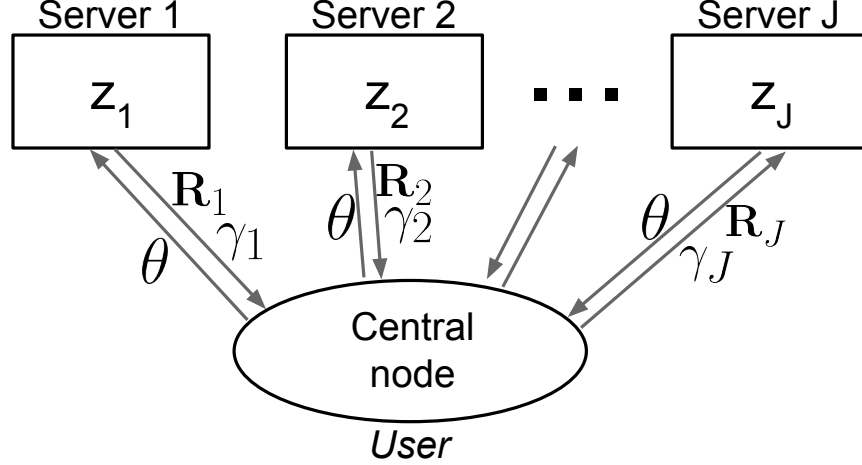


Figure 1: An illustration of the set-up for distributed data with a central node and data stored at J servers. The quantities to be transferred are described in Algorithm 1.

2 Spatial Low-Rank Models

We are interested in making inference on a spatial process $\{y(\mathbf{s}) : \mathbf{s} \in \mathcal{D}\}$, or $y(\cdot)$, on a continuous (non-gridded) domain \mathcal{D} , based on a massive number of measurements, $\mathbf{z}_{1:J} := (\mathbf{z}'_1, \dots, \mathbf{z}'_J)'$, stored on J different servers or data centers, where $\mathbf{z}_j := (z(\mathbf{s}_{j,1}), \dots, z(\mathbf{s}_{j,n_j}))'$ is stored on server j (see Figure 1), and the total number of measurements is given by $n := \sum_{j=1}^J n_j$. Note that the ordering of the servers is completely arbitrary and does not affect the results in any way. We assume that we have additive and spatially independent measurement error, such that

$$z(\mathbf{s}_{j,i}) = y(\mathbf{s}_{j,i}) + \epsilon(\mathbf{s}_{j,i}), \quad (1)$$

for all $i = 1, \dots, n_j$ and $j = 1, \dots, J$, where $\epsilon(\mathbf{s}_i) \sim N(0, v_\epsilon(\mathbf{s}_i))$ is independent of $y(\cdot)$, and the function $v_\epsilon(\cdot)$ is known. In practice, if $v_\epsilon(\cdot)$ is unknown, one can set $v_\epsilon(\cdot) \equiv \sigma_\epsilon^2$, and then estimate σ_ϵ^2 by extrapolating the variogram to the origin (Kang et al., 2009). Because the measurements in (1) are at point level and not on a grid, we assume for simplicity that no two measurement locations coincide exactly.

The true process $y(\cdot)$ is assumed to follow a spatial low-rank model of the form,

$$y(\mathbf{s}) = \mathbf{b}(\mathbf{s})'\boldsymbol{\eta} + \delta(\mathbf{s}), \quad \mathbf{s} \in \mathcal{D}, \quad (2)$$

where $\mathbf{b}(\cdot)$ is a vector of r spatial basis functions with $r \ll n$, $\boldsymbol{\eta} \sim N_r(\boldsymbol{\nu}_0, \mathbf{K}_0)$, and often $\boldsymbol{\nu}_0 = \mathbf{0}$. The fine-scale variation $\delta(\mathbf{s}) \sim N(0, v_\delta(\mathbf{s}))$ is spatially independent and independent of $\boldsymbol{\eta}$. Note that we did not include a spatial trend in (2), as any linear trend of the form $\mathbf{x}(\cdot)'\boldsymbol{\beta}$, where $\mathbf{x}(\cdot)$ is a vector of spatial covariates, can simply be absorbed into $\mathbf{b}(\mathbf{s})'\boldsymbol{\eta}$ if we assign a normal prior distribution to $\boldsymbol{\beta}$. (Then the corresponding elements of $\boldsymbol{\nu}_0$ will typically be nonzero.)

Low-rank models of the form (2) are popular because they do not assume stationarity, and exact spatial predictions can be obtained in a number of flops that is linear in the number of measurements, hence offering excellent scalability for massive datasets. Many

widely used classes of spatial models are of the form (2), such as the spatial random effects model (Cressie and Johannesson, 2008), discretized convolution models (e.g., Higdon, 1998; Calder, 2007; Lemos and Sansó, 2009), and the predictive process (Banerjee et al., 2008; Finley et al., 2009). Basis functions that have been used in (2) include empirical orthogonal functions (e.g. Mardia et al., 1998; Wikle and Cressie, 1999), Fourier basis functions (e.g., Xu et al., 2005), W-wavelets (e.g., Shi and Cressie, 2007; Cressie et al., 2010; Kang and Cressie, 2011), and bisquare functions (e.g., Cressie and Johannesson, 2008; Katzfuss and Cressie, 2011, 2012).

Note that our methodology described in the following sections is applicable to any of these parameterizations of (2), and we do not advocate for a particular model over others. Hence, we work with the general class of spatial low-rank models in (2), and we only assume that there is some parameter vector, $\boldsymbol{\theta}$, that determines $\mathbf{b}(\cdot)$, \mathbf{K}_0 , and $v_\delta(\cdot)$.

3 Distributed Spatial Inference — Main Algorithm

We will now discuss how to obtain the posterior distribution $[\boldsymbol{\eta}|\mathbf{z}_{1:J}]$ by performing parallel computations at each server j that use only the local data \mathbf{z}_j . Throughout this section, we will treat the parameter vector $\boldsymbol{\theta}$ as fixed, with parameter inference to be discussed in Section 4.

First, note that our spatial low-rank model (1)–(2) can be written as a state-space model with state vector $\boldsymbol{\eta}$:

$$\mathbf{z}_j = \mathbf{B}_j \boldsymbol{\eta} + \boldsymbol{\xi}_j, \quad j = 1, \dots, J, \quad (3)$$

where $\boldsymbol{\eta} \sim N(\boldsymbol{\nu}_0, \mathbf{K}_0)$, $\boldsymbol{\xi}_j \stackrel{\text{ind.}}{\sim} N_{n_j}(\mathbf{0}, \mathbf{V}_j)$, $j = 1, \dots, J$, and the local quantities at server j are \mathbf{z}_j , $\mathbf{B}_j := (\mathbf{b}(\mathbf{s}_{j,1}), \dots, \mathbf{b}(\mathbf{s}_{j,n_j}))'$, and $\mathbf{V}_j := \text{diag}(v_\delta(\mathbf{s}_{j,1}) + v_\epsilon(\mathbf{s}_{j,1}), \dots, v_\delta(\mathbf{s}_{j,n_j}) + v_\epsilon(\mathbf{s}_{j,n_j}))$.

It is easy to see that the posterior distribution of $\boldsymbol{\eta}$ given the data at all servers is multivariate normal, $\boldsymbol{\eta}|\mathbf{z}_{1:J} \sim N_r(\boldsymbol{\nu}_z, \mathbf{K}_z)$. The key to our distributed algorithms is that $\boldsymbol{\nu}_z := E(\boldsymbol{\eta}|\mathbf{z}_{1:J})$ and $\mathbf{K}_z := \text{var}(\boldsymbol{\eta}|\mathbf{z}_{1:J})$ can be written as:

$$\begin{aligned} \mathbf{K}_z^{-1} &= \mathbf{K}_0^{-1} + \sum_{j=1}^J \mathbf{R}_j, \\ \boldsymbol{\nu}_z &= \mathbf{K}_z(\mathbf{K}_0^{-1} \boldsymbol{\nu}_0 + \sum_{j=1}^J \boldsymbol{\gamma}_j), \end{aligned} \quad (4)$$

where $\mathbf{R}_j := \mathbf{B}_j' \mathbf{V}_j^{-1} \mathbf{B}_j$ and $\boldsymbol{\gamma}_j := \mathbf{B}_j' \mathbf{V}_j^{-1} \mathbf{z}_j$ are the only quantities that explicitly depend on the data and their spatial locations. The result in (4) can be shown by applying the decentralized Kalman filter (Rao et al., 1993) to the state-space model (3).

This implies the following parallel algorithm to obtain the posterior distribution of $\boldsymbol{\eta}$:

Algorithm 1: Distributed Spatial Inference

1. Do the following *in parallel* for $j = 1, \dots, J$:
 - (a) Move $\boldsymbol{\theta}$ to server j (where data \mathbf{z}_j is stored) and create the matrices \mathbf{B}_j and \mathbf{V}_j .

- (b) At server j , calculate $\mathbf{R}_j = \mathbf{B}_j' \mathbf{V}_j^{-1} \mathbf{B}_j$ and $\boldsymbol{\gamma}_j = \mathbf{B}_j' \mathbf{V}_j^{-1} \mathbf{z}_j$.
 - (c) Transfer the $r \times r$ matrix \mathbf{R}_j and the $r \times 1$ vector $\boldsymbol{\gamma}_j$ back to the central node.
2. At the central node, calculate $\mathbf{K}_z^{-1} = \mathbf{K}_0^{-1} + \sum_{j=1}^J \mathbf{R}_j$ and $\boldsymbol{\nu}_z = \mathbf{K}_z(\mathbf{K}_0^{-1} \boldsymbol{\nu}_0 + \sum_{j=1}^J \boldsymbol{\gamma}_j)$. The posterior distribution of $\boldsymbol{\eta}$ is given by $\boldsymbol{\eta} | \mathbf{z}_{1:J} \sim N_r(\boldsymbol{\nu}_z, \mathbf{K}_z)$.

Algorithm 1 is illustrated in Figure 1. The computational cost is $\mathcal{O}(n_j r^2)$ at server j and $\mathcal{O}(r^3)$ at the central node, it requires $\mathcal{O}(n_j r)$ memory at server j , and we need to move only the $r(r/2 + 3/2)$ unique elements in \mathbf{R}_j and $\boldsymbol{\gamma}_j$ from each server. Compare this to a non-distributed algorithm that has computational cost $\mathcal{O}(nr^2)$, requires $\mathcal{O}(nr)$ memory, and requires moving the n measurements (plus their spatial coordinates) to the central node. In summary, Algorithm 1 has computational cost that is linear in each n_j , the communication cost does not depend on n at all, and hence it is scalable for massive distributed datasets.

3.1 Reducing Communication Via Sparsity

The required amount of communication for Algorithm 1 can be reduced further if the basis-function matrices \mathbf{B}_j sparse, resulting in sparse \mathbf{R}_j . Sparsity can be achieved, for example, by assuming compactly supported basis functions in (2) or by taking the predictive-process approach (Banerjee et al., 2008) with a compactly supported parent covariance function. We will now discuss the latter case in more detail.

Assume a set of knots, $\mathcal{W} := \{\mathbf{w}_1, \dots, \mathbf{w}_r\}$, and a parent covariance function

$$C(\mathbf{s}_1, \mathbf{s}_2) = \sigma(\mathbf{s}_1)\sigma(\mathbf{s}_2)\rho(\mathbf{s}_1, \mathbf{s}_2), \quad \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{D},$$

where ρ is a correlation function. Then the predictive process can be written in the form (2) with

$$\mathbf{b}(\mathbf{s}) := \sigma(\mathbf{s}) (\rho(\mathbf{s}, \mathbf{w}_1), \dots, \rho(\mathbf{s}, \mathbf{w}_r))', \quad \mathbf{s} \in \mathcal{D}, \quad (5)$$

and $\mathbf{K}_0^{-1} = (\rho(\mathbf{w}_i, \mathbf{w}_j))_{i,j=1,\dots,r}$ (see, e.g., Katzfuss, 2013).

Now, if C is compactly supported with range h , then the (l, m) th element of the matrix $\mathbf{R}_j = \mathbf{B}_j' \mathbf{V}_j^{-1} \mathbf{B}_j$ in (4) can only be nonzero if $\|\mathbf{w}_l - \mathbf{w}_m\| < 2h$. Hence, if for a given set of knots, at most v other knots are within a distance of $2h$ of any knot, at most $r(v/2 + 2)$ numbers (including $\boldsymbol{\gamma}_j$) need to be transferred from each server.

4 Parameter Inference

So far, we have treated the parameter vector $\boldsymbol{\theta}$ (containing the parameters determining $\mathbf{b}(\cdot)$, \mathbf{K}_0 , and $v_\delta(\cdot)$) as fixed and known. In practice, of course, this is usually not the case. Fortunately, several commonly used inference approaches can be implemented in a distributed and parallel fashion by extending Algorithm 1 (while still producing the same results as in the traditional, non-distributed setting).

4.1 Parsimonious Parameterizations

If the parameter vector $\boldsymbol{\theta}$ is of low dimension (e.g., there are only three parameters in the predictive-process model in (5) with a Matérn parent covariance function), and estimates or posterior distributions of the parameters are not available in closed form, standard numerical likelihood-based inference is one possibility for parameter inference.

As shown in Appendix A, the likelihood (up to a normalization constant) for the spatial low-rank model in Section 2 can be written as,

$$\begin{aligned} -2 \log L(\boldsymbol{\theta}) &:= -2 \log[\mathbf{z}_{1:J} | \boldsymbol{\theta}] = \\ &= -\log |\mathbf{K}_0^{-1}| + \boldsymbol{\nu}_0' \mathbf{K}_0^{-1} \boldsymbol{\nu}_0 \\ &+ \log |\mathbf{K}_z^{-1}| - \boldsymbol{\nu}_z' \mathbf{K}_z^{-1} \boldsymbol{\nu}_z + \sum_{j=1}^J a_j, \end{aligned} \quad (6)$$

where $a_j := \log |\mathbf{V}_j| + \mathbf{z}_j' \mathbf{V}_j^{-1} \mathbf{z}_j$. This allows us to carry out frequentist inference using numerical maximization of the likelihood, or Bayesian inference using the Metropolis-Hasting algorithm for distributed data. Each iteration of such a parameter-inference procedure consists of carrying out Algorithm 1 (with the addition of calculating a_j at server j and moving this scalar quantity to the central node), combining the results to evaluate the likelihood (6) at the central node, updating the parameters $\boldsymbol{\theta}$, and sending out the new value of $\boldsymbol{\theta}$ to the servers. This results in a sequential algorithm, for which the (major) calculations at each iteration can be carried out in parallel.

To avoid servers being idle in such a sequential algorithm, we recommend instead the use of an importance or particle sampler. Any of the various such algorithms proposed in the literature can be carried out in the distributed context (with the exact same results), by evaluating the likelihood as in (6). Here is an example of such an algorithm:

Algorithm 2: Distributed Importance Sampler

1. Generate a number of parameter vectors or particles, $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(M)}$, from a suitably chosen proposal distribution, $q(\boldsymbol{\theta})$.
2. Do the following *in parallel* for $j = 1, \dots, J$ and $m = 1, \dots, M$:
 - (a) Move $\boldsymbol{\theta}^{(m)}$ to server j and create the matrices $\mathbf{B}_j^{(m)}$ and $\mathbf{V}_j^{(m)}$.
 - (b) Calculate

$$\begin{aligned} \mathbf{R}_j^{(m)} &= \mathbf{B}_j^{(m)'} (\mathbf{V}_j^{(m)})^{-1} \mathbf{B}_j^{(m)} \\ \boldsymbol{\gamma}_j^{(m)} &= \mathbf{B}_j^{(m)'} (\mathbf{V}_j^{(m)})^{-1} \mathbf{z}_j \\ a_j^{(m)} &= \log |\mathbf{V}_j^{(m)}| + \mathbf{z}_j' (\mathbf{V}_j^{(m)})^{-1} \mathbf{z}_j. \end{aligned}$$

- (c) Transfer $\mathbf{R}_j^{(m)}$, $\boldsymbol{\gamma}_j^{(m)}$, and $a_j^{(m)}$ back to the central node.

3. At the central node, for $m = 1, \dots, M$, calculate $(\mathbf{K}_z^{(m)})^{-1} = (\mathbf{K}_0^{(m)})^{-1} + \sum_{j=1}^J \mathbf{R}_j^{(m)}$, $\boldsymbol{\nu}_z^{(m)} = \mathbf{K}_z^{(m)}((\mathbf{K}_0^{(m)})^{-1}\boldsymbol{\nu}_0^{(m)} + \sum_{j=1}^J \boldsymbol{\gamma}_j^{(m)})$, and

$$\begin{aligned}
& -2 \log L(\boldsymbol{\theta}^{(m)}) = \\
& -\log |(\mathbf{K}_0^{(m)})^{-1}| + \boldsymbol{\nu}_0^{(m)'} (\mathbf{K}_0^{(m)})^{-1} \boldsymbol{\nu}_0^{(m)} \\
& + \log |(\mathbf{K}_z^{(m)})^{-1}| - \boldsymbol{\nu}_z^{(m)'} (\mathbf{K}_z^{(m)})^{-1} \boldsymbol{\nu}_z^{(m)} \\
& + \sum_{j=1}^J a_j^{(m)}.
\end{aligned}$$

4. The particle approximation of the posterior distribution of $\boldsymbol{\theta}$ takes on the value $\boldsymbol{\theta}^{(m)}$ with probability $w^{(m)} \propto p(\boldsymbol{\theta}^{(m)})L(\boldsymbol{\theta}^{(m)})/q(\boldsymbol{\theta}^{(m)})$ for $m = 1, \dots, M$, where $p(\boldsymbol{\theta})$ is the prior distribution of the parameters.

The advantage of this parameter-inference approach is that we can carry out calculations for the likelihood evaluations for all particles completely in parallel at all servers (while getting the same results as in the traditional, non-distributed setting).

4.2 Spatial Random Effects Model

In the standard spatial random effects model (Cressie and Johannesson, 2008; Katzfuss and Cressie, 2009; Kang and Cressie, 2011), the basis functions are fixed (i.e., they do not depend on unknown parameters), \mathbf{K}_0 is a general covariance matrix (i.e., it contains $r(r+1)/2$ parameters), and often $v_\delta(\cdot) \equiv \sigma_\delta^2$. If we also assume $v_\epsilon(\cdot) \equiv \sigma_\epsilon^2$ (or we have transformed the data such that these assumptions hold), we have $\mathbf{V}_j^{-1} = \frac{1}{\sigma_\delta^2 + \sigma_\epsilon^2} \mathbf{I}_{n_j}$, and so $\mathbf{R}_j = \frac{1}{\sigma_\delta^2 + \sigma_\epsilon^2} \mathbf{B}_j' \mathbf{B}_j$ and $\boldsymbol{\gamma}_j = \frac{1}{\sigma_\delta^2 + \sigma_\epsilon^2} \mathbf{B}_j' \mathbf{z}_j$. Since the \mathbf{B}_j in the spatial random effects model are fixed, all that is required for inference on $\boldsymbol{\eta}$ in Algorithm 1 from server j are the fixed quantities $\mathbf{B}_j' \mathbf{z}_j$ and $\mathbf{B}_j' \mathbf{B}_j$, making multiple passes over the servers for parameter inference unnecessary. The only additional information required from server j for evaluating the likelihood (6) is n_j and $\mathbf{z}_j' \mathbf{z}_j$.

If the basis functions do contain unknown parameters, or $v_\epsilon(\cdot)$ is not constant, maximum likelihood estimates can be obtained by deriving a distributed version of the expectation-maximization algorithm of Katzfuss and Cressie (2009, 2011). Each step of the resulting algorithm consists of carrying out Algorithm 1, and then updating the estimates of \mathbf{K}_0 and σ_δ^2 as $\mathbf{K}_0^u = \mathbf{K}_z + \boldsymbol{\eta}_z \boldsymbol{\eta}_z'$ and

$$(\sigma_\delta^2)^u = \sigma_\delta^2 + \sum_{j=1}^J (\sigma_\delta^4/n_j) (\|\mathbf{V}_j^{-1}(\mathbf{z}_j - \mathbf{B}_j \boldsymbol{\nu}_z)\|^2 - \text{tr}(\boldsymbol{\Omega}_j^{-1})), \quad (7)$$

respectively, where $\boldsymbol{\Omega}_j := \mathbf{B}_j \mathbf{K}_z \mathbf{B}_j' + \mathbf{V}_j$. The expression for $(\sigma_\delta^2)^u$ in (7) can be derived by obtaining $[\boldsymbol{\delta}_j | \boldsymbol{\eta}, \mathbf{z}_{1:J}]$ and then applying the laws of total expectation and total variance.

By assuming conjugate prior distributions (i.e., an inverse-Wishart distribution for \mathbf{K}_0 and an inverse-Gamma distribution for σ_δ^2), Bayesian inference using a Gibbs sampler is also possible.

5 Spatial Prediction

The goal in spatial statistics is typically to make spatial predictions of $y(\cdot)$ at a set of prediction locations, $\mathbf{s}_1^P, \dots, \mathbf{s}_{n_P}^P$, based on all data $\mathbf{z}_{1:J}$, which in technical terms amounts to finding the posterior predictive distribution $[\mathbf{y}^P | \mathbf{z}_{1:J}]$, where $\mathbf{y}^P := (y(\mathbf{s}_1^P), \dots, y(\mathbf{s}_{n_P}^P))'$. Note that prediction can be carried out separately, after parameter inference has been completed, and so it suffices to obtain the predictive distribution for the final parameter estimates in a frequentist procedure, or for thinned MCMC samples or for particles with nonzero weight in a Bayesian context.

Because we can write

$$\mathbf{y}^P = \mathbf{B}^P \boldsymbol{\eta} + \boldsymbol{\delta}^P, \quad (8)$$

where $\mathbf{B}^P := (\mathbf{b}(\mathbf{s}_1^P), \dots, \mathbf{b}(\mathbf{s}_{n_P}^P))'$ and $\boldsymbol{\delta}^P := (\delta(\mathbf{s}_1^P), \dots, \delta(\mathbf{s}_{n_P}^P))'$, the desired predictive distribution is determined by the joint posterior distribution $[\boldsymbol{\eta}, \boldsymbol{\delta}^P | \mathbf{z}_{1:J}]$.

First, assume that none of the prediction locations exactly coincide with any of the observed locations. This is a reasonable assumption when measurements have point support on a continuous spatial domain, as we have assumed throughout this manuscript. Then it is easy to see that $\boldsymbol{\delta}^P | \mathbf{z}_{1:J} \sim N(\mathbf{0}, \mathbf{V}_\delta^P)$, with $\mathbf{V}_\delta^P := \text{diag}\{v_\delta(\mathbf{s}_1^P), \dots, v_\delta(\mathbf{s}_{n_P}^P)\}$, is conditionally independent of $\boldsymbol{\eta}$ given $\mathbf{z}_{1:J}$. Therefore, spatial prediction reduces to obtaining $\boldsymbol{\nu}_z$ and \mathbf{K}_z using Algorithm 1, and then calculating

$$\mathbf{y}^P | \mathbf{z}_{1:J} \sim N(\mathbf{B}^P \boldsymbol{\nu}_z, \mathbf{B}^P \mathbf{K}_z \mathbf{B}^{P'} + \mathbf{V}_\delta^P) \quad (9)$$

at the central node.

Appendix B describes how to do spatial prediction when a small number of the observed locations coincide with the desired prediction locations.

6 Spatio-Temporal Inference

To extend our results to the spatio-temporal case, we consider a spatio-temporal low-rank model in discrete time of the form,

$$y_t(\mathbf{s}) = \mathbf{b}_t(\mathbf{s})' \boldsymbol{\eta}_t + \delta_t(\mathbf{s}), \quad \mathbf{s} \in \mathcal{D}; \quad t = 1, 2, \dots,$$

where $\delta_t(\cdot)$ is assumed to be independent over space and time, and the temporal evolution of the low-rank component is given by a vector autoregressive model of order one,

$$\boldsymbol{\eta}_t | \boldsymbol{\eta}_{t-1}, \boldsymbol{\eta}_{t-2}, \dots \sim N_r(\mathbf{H}_t \boldsymbol{\eta}_{t-1}, \mathbf{U}_t), \quad t = 1, 2, \dots,$$

with initial state $\boldsymbol{\eta}_0 \sim N_r(\boldsymbol{\nu}_{0,0}, \mathbf{K}_{0,0})$. The data at server j at time t are given by $\mathbf{z}_{j,t} := (z_t(\mathbf{s}_{1,j,t}), \dots, z_t(\mathbf{s}_{n_{j,t},j,t}))'$, with

$$z_t(\mathbf{s}_{i,j,t}) = y_t(\mathbf{s}_{i,j,t}) + \epsilon_t(\mathbf{s}_{i,j,t}),$$

for all $i = 1, \dots, n_{j,t}$, $j = 1, \dots, J$, and $t = 1, 2, \dots$, where $\epsilon_t(\mathbf{s}_{i,j,t}) \sim N(0, v_\epsilon(\mathbf{s}_{i,j,t}))$ is independent in space, time, and of $y(\cdot)$. Cressie et al. (2010) called this the spatio-temporal random effects model, but as in the spatial-only case, many different ways of parameterizing such a spatio-temporal low-rank model are possible (see Section 7 for an example).

6.1 Filtering and Smoothing for Fixed Parameters

We first take an on-line, filtering perspective in time, which means that we are interested at time point t in obtaining the filtering distribution $\boldsymbol{\eta}_t | \mathbf{z}_{1:t} \sim N_r(\boldsymbol{\nu}_{t|t}, \mathbf{K}_{t|t})$, where $\mathbf{z}_{1:t}$ denotes the vector of all data collected at the first t time points. Obtaining $\boldsymbol{\nu}_{t|t}$ and $\mathbf{K}_{t|t}$ can again be achieved using the decentralized Kalman filter (Rao et al., 1993). It requires two nested filters, where each “outer” filtering step over time essentially consists of an “inner” information filter over servers as in (4):

Algorithm 3: Distributed Spatio-Temporal Filtering

1. For $t = 0$, initialize the algorithm with a prior mean, $\boldsymbol{\nu}_{0|0}$, and a prior covariance matrix, $\mathbf{K}_{0|0}$.
2. At time $t = 1, 2, \dots$, once the new data $\mathbf{z}_{1,t}, \dots, \mathbf{z}_{J,t}$ become available:
 - (a) Do the following *in parallel* for $j = 1, \dots, J$:
 - i. Move $\boldsymbol{\theta}$ to server j and create the matrices $\mathbf{B}_{j,t}$ and $\mathbf{V}_{j,t}$ based on the observed locations at time t .
 - ii. At server j , calculate $\mathbf{R}_{j,t} = \mathbf{B}_{j,t}' \mathbf{V}_{j,t}^{-1} \mathbf{B}_{j,t}$ and $\boldsymbol{\gamma}_{j,t} = \mathbf{B}_{j,t}' \mathbf{V}_{j,t}^{-1} \mathbf{z}_{j,t}$.
 - iii. Transfer $\mathbf{R}_{j,t}$ and $\boldsymbol{\gamma}_{j,t}$ back to the central node.
 - (b) At the central node, calculate the forecast quantities $\boldsymbol{\nu}_{t|t-1} := \mathbf{H}_t \boldsymbol{\nu}_{t-1|t-1}$, $\mathbf{K}_{t|t-1} := \mathbf{H}_t \mathbf{K}_{t-1|t-1} \mathbf{H}_t' + \mathbf{U}_t$, and then the filtering quantities $\mathbf{K}_{t|t}^{-1} = \mathbf{K}_{t|t-1}^{-1} + \sum_{j=1}^J \mathbf{R}_{j,t}$ and $\boldsymbol{\nu}_{t|t} = \mathbf{K}_{t|t} (\mathbf{K}_{t|t-1}^{-1} \boldsymbol{\nu}_{t|t-1} + \sum_{j=1}^J \boldsymbol{\gamma}_{j,t})$. We have $\boldsymbol{\eta}_t | \mathbf{z}_{1:t} \sim N_r(\boldsymbol{\nu}_{t|t}, \mathbf{K}_{t|t})$.

In some applications, retrospective smoothing inference based on data collected at T time points might be of interest. Obtaining the smoothing distribution $\boldsymbol{\eta}_t | \mathbf{z}_{1:T} \sim N_r(\boldsymbol{\nu}_{t|T}, \mathbf{K}_{t|T})$ for $t = 1, \dots, T$, requires forward-filtering using Algorithm 3 and then backward-smoothing at the central node by calculating iteratively for $t = T - 1, T - 2, \dots, 1$:

$$\begin{aligned} \boldsymbol{\eta}_{t|T} &= \boldsymbol{\eta}_{t|t} + \mathbf{J}_t (\boldsymbol{\eta}_{t+1|T} - \boldsymbol{\eta}_{t+1|t}), \\ \mathbf{K}_{t|T} &= \mathbf{K}_{t|t} + \mathbf{J}_t (\mathbf{K}_{t+1|T} - \mathbf{K}_{t+1|t}) \mathbf{J}_t', \end{aligned}$$

where $\mathbf{J}_t := \mathbf{K}_{t|t} \mathbf{H}_{t+1}' \mathbf{K}_{t+1|t}^{-1}$ (see, e.g., Cressie et al., 2010, p. 732, for more details). Also, note that in the smoothing context, it is not actually necessary to “consolidate” the information at the end of each time point as in Step 2(b) of Algorithm 3 before moving on to the next time point; instead, we can calculate $\mathbf{R}_{j,1}, \dots, \mathbf{R}_{j,T}$ and $\boldsymbol{\gamma}_{j,1}, \dots, \boldsymbol{\gamma}_{j,T}$ at each server j , and then directly calculate $\mathbf{K}_{T|T}$ and $\boldsymbol{\nu}_{T|T}$ at the central node.

Because $\delta_t(\cdot)$ is *a priori* independent over time, spatial prediction for each t in the filtering and smoothing context can be carried out as described in Section 5 using the filtering or smoothing distribution of $\boldsymbol{\eta}_t$ (i.e., $\boldsymbol{\nu}_{t|t}, \mathbf{K}_{t|t}$ or $\boldsymbol{\nu}_{t|T}, \mathbf{K}_{t|T}$, respectively).

6.2 Spatio-Temporal Parameter Inference

In the filtering context, parameter inference at time point t is typically based on the filtering likelihood,

$$\begin{aligned} -2 \log[\mathbf{z}_t | \mathbf{z}_{1:t-1}, \boldsymbol{\theta}] = & \\ & -\log |\mathbf{K}_{t|t-1}^{-1}| + \boldsymbol{\nu}'_{t|t-1} \mathbf{K}_{t|t-1}^{-1} \boldsymbol{\nu}_{t|t-1} \\ & + \log |\mathbf{K}_{t|t}^{-1}| - \boldsymbol{\nu}'_{t|t} \mathbf{K}_{t|t}^{-1} \boldsymbol{\nu}_{t|t} + \sum_{j=1}^J a_{j,t}, \end{aligned} \quad (10)$$

where $a_{j,t} := \log |\mathbf{V}_{j,t}| + \mathbf{z}'_{j,t} \mathbf{V}_{j,t}^{-1} \mathbf{z}_{j,t}$. This expression of the likelihood can be derived similarly as in the spatial-only case described in Appendix A. If there are a small number of unknown parameters in the spatio-temporal low-rank model, we again advocate the use of a particle-filtering approach for parameter estimation. Sequential importance resampling (Gordon et al., 1993) is a natural inference procedure for on-line inference over time. With distributed data, it can be carried out using a straightforward combination of Algorithms 2 and 3. After the particles at time t have been weighted as described in Algorithm 2 using the likelihood (10), we then resample M particles according to their weights to obtain the particles for time $t + 1$.

In a smoothing context, parameter inference is based on the likelihood of all data, $[\mathbf{z}_{1:T} | \boldsymbol{\theta}] = \prod_{t=1}^T [\mathbf{z}_t | \mathbf{z}_{1:t-1}, \boldsymbol{\theta}]$, where $[\mathbf{z}_t | \mathbf{z}_{1:t-1}, \boldsymbol{\theta}]$ is given in (10).

7 Application: Total Precipitable Water Measured by Three Sensor Systems

We applied our methodology to hourly measurements from three sensor systems to obtain spatio-temporal filtering inference on an atmospheric variable called total precipitable water. The sensor systems are ground-based GPS, the Geostationary Operational Environmental Satellite (GOES) infrared sounders, and Microwave Integrated Retrieval System (MIRS) satellites. These data products are retrieved and stored at different data centers and feature varying spatial coverage and precision. The measurement-error standard deviations are 0.75 mm, 2 mm, and 4.5 mm, respectively, and so the function $v_\epsilon(\cdot)$ from (1) varies by server (i.e., by j) but not over space. Since March 2009, an operational blended multisensor water vapor product based on these three sensor systems has been produced by the National Environmental Satellite, Data, and Information Service of NOAA (Kidder and Jones, 2007; Forsythe et al., 2012). This product is sent to National Weather Service offices, where it is used by forecasters to track the movement of water vapor in the atmosphere and to detect antecedent conditions for heavy precipitation. The operational product is created by overlaying the existing field with the latest available data, which can lead to unphysical features in the form of abrupt boundaries. The goal of our analysis was to illustrate our methodology using a simple version of a spatio-temporal low-rank model, and to create spatially more coherent predictive maps with associated uncertainties based on data from all three systems, without having to transfer the data to a central processor.

We consider here a dataset consisting of a total of 3,351,860 measurements assumed to be collected at point-level support in January 2011 over a period of 47 hours by the

Figure 2: Top row: Hourly observations of total precipitable water by (a) the GPS system, (b) GOES infrared sounders, and (c) MIRS over the larger continental United States in January 2011. Bottom row: Filtering (d) posterior means and (e) posterior standard deviations of total precipitable water based on all three data products. All units are in millimeters.

three sensor systems over a spatial domain covering the United States. The top row of Figure 2 shows the three sensor data products. We made filtering inference using sequential importance resampling as described in Section 6.2 based on a spatio-temporal low-rank model, parameterized by a predictive-process approach inspired by Finley et al. (2012), with an isotropic Matérn parent covariance function (e.g., Stein, 1999, p. 50). Specifically, we assumed the model in Section 6 with $v_{\delta,t}(\cdot) \equiv \sigma_{\delta,t}^2$, $\mathbf{H}_t = \alpha_t \mathbf{I}_r$,

$$\begin{aligned}\mathbf{K}_{0,0}^{-1} &= (\rho(\mathbf{w}_i, \mathbf{w}_j | \boldsymbol{\theta}_0))_{i,j=1,\dots,r} \\ \mathbf{U}_t^{-1} &= (1 - \alpha_t^2)^{-1} (\rho(\mathbf{w}_i, \mathbf{w}_j | \boldsymbol{\theta}_t))_{i,j=1,\dots,r} \\ \mathbf{b}_t(\mathbf{s}) &= \sigma_t(\mathbf{s}) \left(\rho(\mathbf{s}, \mathbf{w}_1 | \boldsymbol{\theta}_t), \dots, \rho(\mathbf{s}, \mathbf{w}_r | \boldsymbol{\theta}_t) \right)', \mathbf{s} \in \mathcal{D},\end{aligned}$$

where

$$\rho(\mathbf{s}_1, \mathbf{s}_2 | \boldsymbol{\theta}_t) = (2h\sqrt{v_t})_t^v \mathcal{K}_v(2h\sqrt{v_t}) 2^{1-v_t} / \Gamma(v_t),$$

with $h = \|\mathbf{s}_1 - \mathbf{s}_2\|/\kappa_t$, and the set of knots, $\mathcal{W} := \{\mathbf{w}_1, \dots, \mathbf{w}_{84}\}$, was a regular $5^\circ \times 5^\circ$ latitude/longitude grid over the domain. The trend consisted of an intercept term with a Gaussian random-walk prior with initial value 13.2 and variance 15.9 and was absorbed into the basis-function vector. While we chose this relatively simple model here for illustration, we would like to reiterate that neither the communication cost nor the computational complexity

of the algorithm changes if a more elaborate parameterization of the general spatio-temporal low-rank model in Section 6 is chosen.

The prior distribution of the parameter vector

$$\boldsymbol{\theta}_t = (\Phi^{-1}(\alpha_t), \log(\sigma_t), \Phi^{-1}(v_t/2), \log(\kappa_t), \log(\sigma_{\delta,t}^2))'$$

was also taken to be a Gaussian random walk with initial values $\alpha_0 = 0.8$, $\sigma_0 = 5$, $v_0 = 1.25$, $\kappa_0 = 15$, $\sigma_{\delta,0}^2 = 0.5$ and covariance matrix $0.1 \times \mathbf{I}_5$. Here, α_t determines the strength of the temporal dependence, while the smoothness parameter $v_t \in (0, 2)$ and the scale parameter κ_t determine the strength of the spatial dependence.

Our sequential importance resampling algorithm as described in Section 6.2 (using the prior distribution as the proposal distribution for simplicity) had $P = 1000$ particles, and we only carried out spatial prediction for particles with nonzero weight ($> .0001$) for fast computation. The resulting filtering posterior means and posterior standard deviations for total precipitable water for time period 1 (i.e., $t = 1$) on a regular $0.5^\circ \times 0.5^\circ$ latitude/longitude grid of size 6,283 are shown in the bottom row of Figure 2. We were able to calculate the (fully Bayesian) filtering distribution based on the 3,351,860 measurements collected by the three sensor systems over 47 hours in only 19 hours on a standard laptop computer (MacBook Pro with an Intel quad-core 2.6 GHz i7 processor and 8 GB of memory).

8 Conclusions and Future Work

As datasets are becoming larger, so is the cost of moving them to a central computer for analysis, necessitating algorithms designed to work on distributed data that keep analysis operations as close to the stored data as possible. We showed how distributed spatial inference, including likelihood-based parameter inference, can be carried out in a computationally feasible way for massive distributed datasets under the assumption of a low-rank model, while producing the same results as traditional, non-distributed inference. Our approach is scalable in that the computational cost is linear in n (the number of measurements) and the communication cost does not depend on n at all. The inference, especially when done based on a particle sampler, is also “embarrassingly parallel,” allowing a divide-and-conquer analysis of massive spatial data with little communication overhead. In addition, if the selected low-rank model has fixed basis functions that do not depend on parameters (see Section 4.2), our methodology can be used for data reduction in situations where it is not possible to store all measurements.

After extending the results to the spatio-temporal case, we demonstrated the applicability of our model to massive real-world data in Section 7, and showed that we can obtain sensible results in a fast manner. However, getting the best possible results for this particular application is part of ongoing research and will likely require a more refined and complicated model.

The methodology described in this article can be extended to the full-scale approximation of Sang et al. (2011), where the fine-scale variation is assumed to be dependent within subregions of the spatial domain, resulting in nondiagonal \mathbf{V}_j , but this will be explored further in future work.

Another natural extension of our methodology is to the increasingly important multivariate data-fusion case involving inference on multiple processes based on data from multiple measuring instruments. Multivariate analysis can in principle be carried out as described here by stacking the basis function weights for the individual processes into one big vector $\boldsymbol{\eta}$ (see, e.g., Nguyen et al., 2012, 2014), but it will likely require more complicated inference on $\delta(\cdot)$ due to different instrument footprints and overlaps. While the combined size of the low-rank components for multiple processes will become prohibitive in highly multivariate settings, the hope is that the processes can be written as linear combinations of a smaller number of processes.

Acknowledgments

This material was based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Hammerling’s research also had partial support from the NSF Research Network on Statistics in the Atmosphere and Ocean Sciences (STATMOS) through grant DMS-1106862. We would like to thank Amy Braverman for making us aware of the problem of distributed spatial data; John Forsythe and Stan Kidder for the datasets and helpful advice; Yoichi Shiga for support with preprocessing and visualizing the data; and Andrew Zammit Mangion, Emtiyaz Khan, Kirk Borne, Jessica Matthews, Emily Kang, several anonymous reviewers, and the SAMSI Massive Datasets Environment and Climate working group for helpful comments and discussions.

A Derivation of the Likelihood

We derive here the expression of the likelihood in (6). First, note that $\mathbf{z}_{1:J}|\boldsymbol{\theta} \sim N(\mathbf{B}_{1:J}\boldsymbol{\nu}_0, \boldsymbol{\Sigma}_{1:J})$, where $\mathbf{B}_{1:J} = (\mathbf{B}'_1, \dots, \mathbf{B}'_J)'$, $\boldsymbol{\Sigma}_{1:J} = \mathbf{B}_{1:J}\mathbf{K}_0\mathbf{B}'_{1:J} + \mathbf{V}_{1:J}$, and $\mathbf{V}_{1:J}$ is a blockdiagonal matrix with j th block \mathbf{V}_j . Hence, the likelihood is given by,

$$\begin{aligned} -2\log[\mathbf{z}_{1:J}|\boldsymbol{\theta}] &= \log|\boldsymbol{\Sigma}_{1:J}| \\ &+ (\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0)' \boldsymbol{\Sigma}_{1:J}^{-1} (\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0) - (n/2)\log(2\pi). \end{aligned}$$

Applying a matrix determinant lemma (e.g., Harville, 1997, Thm. 18.1.1), we can write the log determinant as,

$$\begin{aligned} \log|\boldsymbol{\Sigma}_{1:J}| &= \log|\mathbf{V}_{1:J}| + \log|\mathbf{K}_0| \\ &+ \log|\mathbf{B}'_{1:J}\mathbf{V}_{1:J}^{-1}\mathbf{B}_{1:J} + \mathbf{K}_0^{-1}| \\ &= \sum_{j=1}^J \log|\mathbf{V}_j| - \log|\mathbf{K}_0^{-1}| + \log|\mathbf{K}_z^{-1}|. \end{aligned}$$

Further, using the Sherman-Morrison-Woodbury formula (Sherman and Morrison, 1950; Woodbury, 1950; Henderson and Searle, 1981), we can show that $\Sigma_{1:J}^{-1} = \mathbf{V}_{1:J}^{-1} - \mathbf{V}_{1:J}^{-1}\mathbf{B}_{1:J}\mathbf{K}_z\mathbf{B}_{1:J}'\mathbf{V}_{1:J}^{-1}$, and so

$$\begin{aligned}
& (\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0)' \Sigma_{1:J}^{-1} (\mathbf{z}_{1:J} - \mathbf{B}_{1:J}\boldsymbol{\nu}_0) \\
&= \sum_{j=1}^J (\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0)' \mathbf{V}_j^{-1} (\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0) \\
&\quad - \left(\sum_{j=1}^J \mathbf{B}_j' \mathbf{V}_j^{-1} (\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0) \right)' \mathbf{K}_z \\
&\quad \times \left(\sum_{j=1}^J \mathbf{B}_j' \mathbf{V}_j^{-1} (\mathbf{z}_j - \mathbf{B}_j\boldsymbol{\nu}_0) \right) \\
&= \sum_j \mathbf{z}_j' \mathbf{V}_j^{-1} \mathbf{z}_j - 2\boldsymbol{\nu}_0' (\mathbf{K}_z^{-1} \boldsymbol{\nu}_z - \mathbf{K}_0^{-1} \boldsymbol{\nu}_0) \\
&\quad + \boldsymbol{\nu}_0' (\mathbf{K}_z^{-1} - \mathbf{K}_0^{-1}) \boldsymbol{\nu}_0 \\
&\quad - \left((\mathbf{K}_z^{-1} \boldsymbol{\nu}_z - \mathbf{K}_0^{-1} \boldsymbol{\nu}_0) - (\mathbf{K}_z^{-1} - \mathbf{K}_0^{-1}) \boldsymbol{\nu}_0 \right)' \mathbf{K}_z \\
&\quad \times \left((\mathbf{K}_z^{-1} \boldsymbol{\nu}_z - \mathbf{K}_0^{-1} \boldsymbol{\nu}_0) - (\mathbf{K}_z^{-1} - \mathbf{K}_0^{-1}) \boldsymbol{\nu}_0 \right) \\
&= \sum_j \mathbf{z}_j' \mathbf{V}_j^{-1} \mathbf{z}_j - 2\boldsymbol{\nu}_0' \mathbf{K}_z^{-1} \boldsymbol{\nu}_z + \boldsymbol{\nu}_0' \mathbf{K}_0^{-1} \boldsymbol{\nu}_0 + \boldsymbol{\nu}_0' \mathbf{K}_z^{-1} \boldsymbol{\nu}_0 \\
&\quad - (\mathbf{K}_z^{-1} \boldsymbol{\nu}_z)' \mathbf{K}_z (\mathbf{K}_z^{-1} \boldsymbol{\nu}_z) - \boldsymbol{\nu}_0' \mathbf{K}_z^{-1} \mathbf{K}_z \mathbf{K}_z^{-1} \boldsymbol{\nu}_0 \\
&\quad + 2(\mathbf{K}_z^{-1} \boldsymbol{\nu}_z)' \mathbf{K}_z \mathbf{K}_z^{-1} \boldsymbol{\nu}_0 \\
&= \sum_j \mathbf{z}_j' \mathbf{V}_j^{-1} \mathbf{z}_j + \boldsymbol{\nu}_0' \mathbf{K}_0^{-1} \boldsymbol{\nu}_0 - \boldsymbol{\nu}_z' \mathbf{K}_z^{-1} \boldsymbol{\nu}_z,
\end{aligned}$$

where $\sum_{j=1}^J \mathbf{B}_j' \mathbf{V}_j^{-1} \mathbf{B}_j = \mathbf{K}_z^{-1} - \mathbf{K}_0^{-1}$ and $\sum_{j=1}^J \mathbf{B}_j' \mathbf{V}_j^{-1} \mathbf{z}_j = \mathbf{K}_z^{-1} \boldsymbol{\nu}_z - \mathbf{K}_0^{-1} \boldsymbol{\nu}_0$ both follow from (4).

B Spatial Prediction When Observed and Prediction Locations Coincide

Here we describe how to do spatial prediction when a small number, q say, of the observed locations are also in the set of desired prediction locations. Define $\boldsymbol{\delta}_{P,O}$ to be the vector of the first q elements of $\boldsymbol{\delta}^P$, which we assume to correspond to the q observed prediction locations, and let \mathbf{P}_j be a sparse $n_j \times q$ matrix with $(\mathbf{P}_j)_{k,l} = I(\mathbf{s}_{j,k} = \mathbf{s}_l^P)$. Again, we write our model in state-space form similar to (3), $\mathbf{z}_j = \tilde{\mathbf{B}}_j \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\xi}}_j$, where $\tilde{\mathbf{B}}_j := (\mathbf{B}_j, \mathbf{P}_j)$, $\tilde{\boldsymbol{\eta}} := (\boldsymbol{\eta}', \boldsymbol{\delta}_{P,O}')' \sim N(\tilde{\boldsymbol{\nu}}_0, \tilde{\mathbf{K}}_0)$, $\tilde{\boldsymbol{\nu}}_0 := (\boldsymbol{\nu}_0', \mathbf{0}_q')$, $\tilde{\mathbf{K}}_0$ is blockdiagonal with first block \mathbf{K}_0 and second block $\text{diag}\{v_\delta(\mathbf{s}_1^P), \dots, v_\delta(\mathbf{s}_q^P)\}$, $\tilde{\boldsymbol{\xi}}_j \sim N_{n_j}(\mathbf{0}, \tilde{\mathbf{V}}_j)$, and $\tilde{\mathbf{V}}_j$ is the same as \mathbf{V}_j except that the i th diagonal element is now $v_\epsilon(\mathbf{s}_{j,i})$ if $\mathbf{s}_{j,i}$ is one of the prediction locations.

The decentralized Kalman filter (Rao et al., 1993) gives $\tilde{\mathbf{K}}_z^{-1} = \tilde{\mathbf{K}}_0^{-1} + \sum_{j=1}^J \tilde{\mathbf{R}}_j$ and $\tilde{\boldsymbol{\nu}}_z = \tilde{\mathbf{K}}_z(\tilde{\mathbf{K}}_0^{-1} \tilde{\boldsymbol{\nu}}_0 + \sum_{j=1}^J \tilde{\boldsymbol{\gamma}}_j)$, where $\tilde{\mathbf{R}}_j := \tilde{\mathbf{B}}_j' \tilde{\mathbf{V}}_j^{-1} \tilde{\mathbf{B}}_j$ and $\tilde{\boldsymbol{\gamma}}_j := \tilde{\mathbf{B}}_j' \tilde{\mathbf{V}}_j^{-1} \mathbf{z}_j$ are the only quantities that need to be calculated at and transferred from server j , which is feasible due to sparsity if q is not too large. The predictive distribution is then given by $\mathbf{y}^P | \mathbf{z}_{1:J} \sim N(\tilde{\mathbf{B}}^P \tilde{\boldsymbol{\nu}}_z, \tilde{\mathbf{B}}^P \tilde{\mathbf{K}}_z \tilde{\mathbf{B}}^{P'} + \tilde{\mathbf{V}}_\delta^P)$, where $\tilde{\mathbf{B}}^P := (\mathbf{B}^P, (\mathbf{I}_q, \mathbf{0})')$ and $\tilde{\mathbf{V}}_\delta^P := \text{diag}\{\mathbf{0}_q', v_\delta(\mathbf{s}_{q+1}^P), \dots, v_\delta(\mathbf{s}_{n_P}^P)\}$.

References

- Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society, Series B*, 70(4):825–848.
- Bevilacqua, M., Gaetan, C., Mateu, J., and Porcu, E. (2012). Estimating space and space-time covariance functions for large data sets: A weighted composite likelihood approach. *Journal of the American Statistical Association*, 107(497):268–280.

- Bradley, J. R., Cressie, N., and Shi, T. (2014). A comparison of spatial predictors when datasets could be very large. *arXiv:1410.7748*.
- Calder, C. A. (2007). Dynamic factor process convolution models for multivariate space-time data with application to air quality assessment. *Environmental and Ecological Statistics*, 14(3):229–247.
- Caragea, P. C. and Smith, R. L. (2007). Asymptotic properties of computationally efficient alternative estimators for a class of multivariate normal models. *Journal of Multivariate Analysis*, 98(7):1417–1440.
- Caragea, P. C. and Smith, R. L. (2008). Approximate Likelihoods for Spatial Processes. Technical Report, University of North Carolina, Chapel Hill, NC.
- Cortés, J. (2009). Distributed kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827.
- Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society, Series B*, 70(1):209–226.
- Cressie, N., Shi, T., and Kang, E. L. (2010). Fixed rank filtering for spatio-temporal data. *Journal of Computational and Graphical Statistics*, 19(3):724–745.
- Curriero, F. and Lele, S. (1999). A composite likelihood approach to semivariogram estimation. *Journal of Agricultural, Biological, and Environmental Statistics*, 4(1):9–28.
- Eidsvik, J., Shaby, B. A., Reich, B. J., Wheeler, M., and Niemi, J. (2014). Estimation and prediction in spatial models with block composite likelihoods using parallel computing. *Journal of Computational and Graphical Statistics*, 23(2):295–315.
- Finley, A., Banerjee, S., and Gelfand, A. E. (2012). Bayesian dynamic modeling for large space-time datasets using Gaussian predictive processes. *Journal of Geographical Systems*, 14:29–47.
- Finley, A. O., Sang, H., Banerjee, S., and Gelfand, A. E. (2009). Improving the performance of predictive process modeling for large datasets. *Computational Statistics & Data Analysis*, 53(8):2873–2884.
- Forsythe, J. M., Dodson, J. B., Partain, P. T., Kidder, S. Q., and Haar, T. H. V. (2012). How total precipitable water vapor anomalies relate to cloud vertical structure. *Journal of Hydrometeorology*, 13(2):709–721.
- Fuller, S. H. and Millett, L. I., editors (2011). *The Future of Computing Performance: Game Over or Next Level?* Committee on Sustaining Growth in Computing Performance; National Research Council, Washington, DC.
- Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113.
- Graham, R. and Cortés, J. (2012). Cooperative adaptive sampling of random fields with partially known covariance. *International Journal of Robust and Nonlinear Control*, 22(5):504–534.
- Harville, D. A. (1997). *Matrix Algebra From a Statistician’s Perspective*. Springer, New York, NY.
- Henderson, H. and Searle, S. (1981). On deriving the inverse of a sum of matrices. *SIAM Review*, 23(1):53–60.
- Higdon, D. (1998). A process-convolution approach to modelling temperatures in the North Atlantic Ocean. *Environmental and Ecological Statistics*, 5(2):173–190.
- Kang, E. L. and Cressie, N. (2011). Bayesian inference for the spatial random effects model. *Journal of the American Statistical Association*, 106(495):972–983.
- Kang, E. L., Liu, D., and Cressie, N. (2009). Statistical analysis of small-area data based on independence, spatial, non-hierarchical, and hierarchical models. *Computational Statistics &*

- Data Analysis*, 53(8):3016–3032.
- Katzfuss, M. (2013). Bayesian nonstationary spatial modeling for very large datasets. *Environmetrics*, 24(3):189–200.
- Katzfuss, M. and Cressie, N. (2009). Maximum likelihood estimation of covariance parameters in the spatial-random-effects model. In *Proceedings of the Joint Statistical Meetings*, pages 3378–3390, Alexandria, VA. American Statistical Association.
- Katzfuss, M. and Cressie, N. (2011). Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets. *Journal of Time Series Analysis*, 32(4):430–446.
- Katzfuss, M. and Cressie, N. (2012). Bayesian hierarchical spatio-temporal smoothing for very large datasets. *Environmetrics*, 23(1):94–107.
- Kidder, S. Q. and Jones, A. S. (2007). A blended satellite total precipitable water product for operational forecasting. *Journal of Atmospheric and Oceanic Technology*, 24(1):74–81.
- Lemos, R. T. and Sansó, B. (2009). A spatio-temporal model for mean, anomaly, and trend fields of North Atlantic sea surface temperature. *Journal of the American Statistical Association*, 104(485):5–18.
- Mardia, K., Goodall, C., Redfern, E., and Alonso, F. (1998). The kriged Kalman filter. *Test*, 7(2):217–282.
- Nguyen, H., Cressie, N., and Braverman, A. (2012). Spatial statistical data fusion for remote sensing applications. *Journal of the American Statistical Association*, 107(499):1004–1018.
- Nguyen, H., Katzfuss, M., Cressie, N., and Braverman, A. (2014). Spatio-temporal data fusion for very large remote sensing datasets. *Technometrics*, 56(2):174–185.
- Rao, B., Durrant-Whyte, H., and Sheen, J. (1993). A fully decentralized multi-sensor system for tracking and surveillance. *The International Journal of Robotics Research*, 12(1):20–44.
- Sang, H., Jun, M., and Huang, J. Z. (2011). Covariance approximation for large multivariate spatial datasets with an application to multiple climate model errors. *Annals of Applied Statistics*, 5(4):2519–2548.
- Sherman, J. and Morrison, W. (1950). Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, 21(1):124–127.
- Shi, T. and Cressie, N. (2007). Global statistical analysis of MISR aerosol data: A massive data product from NASA’s Terra satellite. *Environmetrics*, 18:665–680.
- Shoshani, A., Klasky, S., and Ross, R. (2010). Scientific data management: Challenges and approaches in the extreme scale era. In *Proceedings of the 2010 Scientific Discovery through Advanced Computing (SciDAC) Conference*, number 1, pages 353–366, Chattanooga, TN.
- Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York, NY.
- Stein, M. L. (2014). Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics*, 8:1–19.
- Stein, M. L., Chi, Z., and Welty, L. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B*, 66(2):275–296.
- Vecchia, A. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society, Series B*, 50(2):297–312.
- Wikle, C. K. and Cressie, N. (1999). A dimension-reduced approach to space-time Kalman filtering. *Biometrika*, 86(4):815–829.
- Woodbury, M. (1950). Inverting modified matrices. *Memorandum Report 42, Statistical Research Group, Princeton University*.

- Xu, B., Wike, C. K., and Fox, N. (2005). A kernel-based spatio-temporal dynamical model for nowcasting radar precipitation. *Journal of the American Statistical Association*, 100(472):1133–1144.
- Xu, Y. and Choi, J. (2011). Adaptive sampling for learning gaussian processes using mobile sensor networks. *Sensors*, 11(3):3051–3066.
- Zhang, K. (2013). ISSCC 2013: Memory trends. <http://www.electroiq.com/articles/sst/2013/02/isscc-2013--memory-trends.html>. Accessed June 12, 2013.